Nonlinear Dyn (2022) 110:1613–1631
https://doi.org/10.1007/s11071-022-07695-x

ORIGINAL PAPER

Reduced order dynamical models for complex dynamics in manufacturing and natural systems using machine learning

William Farlessyost · Shweta Singh

Received: 7 February 2022 / Accepted: 27 June 2022 / Published online: 2 August 2022

Abstract Dynamical analysis of manufacturing and natural systems provides critical information about production of manufactured and natural resources, respectively. Current dynamic models for full industrial process plants exist as highly accurate first-principle relationships. However, their integration is computationally intensive and provides no simplified understanding of the underlying mechanisms driving the overall dynamics. Similarly, for natural systems, most dynamical models are first principle based, with high data requirements and low state accuracy. Consequently, lower-order models that may sacrifice accuracy for simplicity and ease of training can prove useful. Yet, there have been few attempts at finding low-order models of chemical manufacturing processes and natural systems, with work focusing on modeling individual mechanisms. We seek to fill this research gap by using a machine learning (ML) approach, SINDy, validated on a soybean-diesel process plant and watershed system. This ML method combines sparse, grey-box modeling with additional nonlinear optimization to identify governing dynamics as ODEs. We find a linear ODE model for the process plant that gives an accurate relation between input and output and selected internal molar flow rates reflective of underlying linear stoichiometric mechanisms and an internal mass balance. For the natural system, we modify the SINDy approach to include the effect of past dynamics on training the model, which gives a nonlinear model for streamflow dynamics. This improves dynamical transitions, but falls short of accurate state estimation. We conclude that the proposed ML approach works well for non-chaotic systems with minimal hysteresis, but is limited when this condition is not met.

Keywords Machine learning · Dynamical equations · Reduced order · Manufacturing systems · Natural systems

W. Farlessyost · S. Singh
Agricultural and Biological Engineering, Purdue University, 225 South University Street, West Lafayette, IN 47907, USA
e-mail: wfarless@purdue.edu

W. Farlessyost
Ecological Sciences and Engineering, Purdue University, 155 S. Grant Street, West Lafayette, IN 47907, USA

S. Singh (✉)
Environmental and Ecological Engineering, Purdue University, 500 Central Drive, City, IN 47907, USA
e-mail: singh294@purdue.edu

1 Introduction

Models that describe, predict, and lend understanding of the dynamics of a system are crucial tools across the scientific fields. Often, models found via system identification or mathematical modeling are broken into white-box, grey-box, and black-box models. Pure white-box models are based wholly on first-principle knowledge [1]. This mechanistic understanding typically drives both the structure and parameter values in a white-box model. Grey-box models refer to models

with first-principle-determined structure where parameters are further tuned based on training data [1]. Thus, model components have physical meaning since the structure of the model is derived from this understanding. Black-box models, however, rely on a generalized model structure that is then further specified through parameter tuning [2]. These models can be highly accurate, but lack a structure with physical interpretation [3].

Yet for certain large, dynamic systems where numerous mechanisms drive state transition, traditional white-box, grey-box, and black-box modeling techniques may be insufficient to develop reduced order, interpretable, and sufficiently accurate models. Take, for example, the models for overall plant dynamics in chemical process engineering. While highly accurate mechanistic models based on first-principle relationships exist for each unit-operation and are used to simulate chemical process dynamics in modeling software like Aspen Plus^{TM}, the overall relationship between internal state variables, inputs, and outputs is difficult to decipher from these aggregate white-box models due to the size and complexity of interactions [4]. Similarly, for dynamic models of natural systems, such as a watershed system, existing models are often grey-box: based on first-principle relationships and mass balances with parameters fit to a specific location based on environmental data [5]. These mechanistic relationships are only accurate when computed with average values over monthly or longer long periods of time [6]. As a result, short-term temporal variations (time scales smaller than a month) in flowrate are not captured, thus limiting their use in applications of model-based control and optimization of water usage habits.

Recently, data-driven approaches for system identification have been proposed to develop dynamical models bypassing the traditional model identification approaches. One approach is the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm, which is a data-driven system identification technique with growing popularity across a variety of fields [7–14].The initial publication by Brunton et al. [15], first introducing the core SINDy algorithm, provides a method of system identification that makes minimal assumptions about the physics of the system or the necessary model structure. Rather, the model structure is defined to be a sum of candidate functions, with coefficients tuned via regression with intermittent thresholding to maximize accuracy and ensure sparsity [16]. In [15], the SINDy algorithm is expanded to include control or forcing inputs to a system. Further, Stender et al. [17] suggest that the SINDy algorithm described in [16] and [15] may be further improved by employing bounded nonlinear optimization of the model coefficients. These previous works have mainly focused on simple systems with minimal complexity and low dimensionality; thus, the validity of approach for large-scale system is not established.

In this paper, we share our results and continue our discussion from [18], where we propose that SINDy can be further applied to recover accurate, interpretable, simplified models for a variety of different complex systems, such as process plants and watershed dynamics. We utilize this SINDy approach with simulated and observational data on these large-scale complex systems. We chose two distinct systems to model dynamics in our study—a chemical transesterification process converting soybean-oil to soybean-diesel and the streamflow dynamics of North Fork Vermilion River. For the chemical system we use mechanistic, simulated data while for the natural system we rely on historical streamflow values and downscaled climate data.

The remaining paper is organized as follows. Section 2 provides background on data-driven system identification and existing dynamical modeling approach for chemical processes and natural processes. Section 3 describes our approach including the mathematical set up of the SINDy algorithm, selection of relevant state variables for our systems, and data generation or selection. Section 4 describes the model recovery results, and Sect. 5 follows with relevant conclusions and discussion.

2 Background

2.1 Background data-driven approach for nonlinear system identification

Since the publication of [16] formalizing the SINDy method, this sparse regression technique has grown popular with researchers across natural sciences and engineering who require differential models to characterize the dynamics of their respective systems. Consequently, SINDy, or SINDy-like methods, have been applied to a wide range of systems. In [7], the SINDy method is used to recover a model that allows the authors to predict the required input force for a given

vibrational response in machine tools that experience cutting forces. The authors of [8] use the SINDy method to find a low-order stochastic model for interaction between the geomagnetic field axial dipole and non-dipole components. In [9], a proposed dimensionless learning approach based on dimensional invariance with the SINDy method is shown to recover dimensionless differential equations with a physically interpretable parameterization. SINDy is used by [10] to recover stochastic differential equations for a vibrational energy harvester. The authors of [11] use the SINDy algorithm to form a model of seismic response in steel-braced beams. To detect load-altering attacks in a power grid, [12] looks for parameterization changes in an online version of the SINDy algorithm. In [19], SINDy is combined with stepwise sparse regression to recover dynamic models for use in control of longitudinal missiles. A modified version of the SINDy method is applied to experimental data in [13] to recover governing dynamic equations of a duffing oscillator. The SINDy method has also been applied to chemical and chemical process systems. In [20], the authors utilize SINDy to recover dynamic equations governing a chemical reaction network. Further, in [14], SINDy is compared against symbolic regression in model recovery of a chemical distillation column. The diverse set of applications discussed previously suggests that the SINDy method may have widespread utility in system identification. However, the ability of this algorithm to recover governing or predictive dynamical equations for a large scale complex system such as overall chemical plant dynamics and watershed has yet to be shown. We next describe the existing approaches and need of reduced order dynamical models in two distinct type of systems—chemical process industries and watersheds.

2.2 Modeling dynamics in chemical process industries

Process industries can be defined as those which apply chemical or mechanical changes to their system inputs to output a product in a continuous or semi-continuous fashion [21]. System identification of these processes is crucial within their respective industries for developing models that can be used for plant design, observation or control. This system identification is typically iterative and data-driven since a priori model structure is often minimal [22]. To limit the amount of disturbance to plant operation, these system identification methods must be "plant-friendly," meaning industries go to great lengths to use data collection experiments that minimizes equipment degradation, plant output deviations, and experiment time [22]. However, because of data confidentiality of plants, the recovered dynamical models from these efforts are rarely published or made publicly available by industry. In our work, we use the SINDy algorithm to recover a dynamical model for a soybean-oil to soybean-diesel transesterification process using simulated time-series data to show how this approach can develop informative models using synthetic data. In particular, we model the dynamic behavior of material flow rates at various points in the process. While models capturing the kinetics of soybean-oil transesterification and the dynamics within the plant reactor [23–25] do exist, plant-wide models that also contain the dynamic relationship between the internal molar flow rates, the output soybean-diesel flow rate, and the input flows is not available. Thus, we aim to develop a reduced order model to capture the overall plant dynamics for the soybean-oil to soybean-diesel process, to demonstrate the application of SINDy for overall chemical plant dynamics.

2.3 Modeling streamflow dynamics in watershed

Streamflow dynamics is an important system to study as it provides insights into water availability over the short and long term. Several standard approaches exist for system identification of watershed models with the earliest simple water-balance models developed in the 1940s [26,27]. These models relate known spatial inputs (precipitation, temperature, etc.) to characteristics that are difficult to measure (evapotranspiration, total-runoff, etc.) [5]. Because full sets of these spatial inputs may not readily available data in all scenarios, models have been developed with various degrees of input resolution [5]. The applicability of these models to various locations and problems is dependent on their timescale and potential accuracy [5]. Their structure can be relatively simple, with only precipitation and temperature used to predict seasonal streamflow with high accuracy [28]. Conversely, these models perform poorly at a finer time-resolution, thus lacking state estimation ability for the system as a whole [6]. Hence, in this work, we demonstrate development of a finer scale model using the data-driven system identification approach. As a representative of a natural

system, we apply the SINDy algorithm to recover a low-level dynamic equation for streamflow dynamics of the North Fork Vermilion River, providing water to the town of Danville, Illinois, via training on historical streamflow and climate data.

3 Materials and methods

3.1 Algorithm for model identification and modifications

3.1.1 SINDy algorithm

The SINDy method, as described in [16], assumes the system in question can be modeled using ordinary differential equation type state equations of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \tag{1}$$

where $\mathbf{x}(t) \in \mathbf{R}^n$ is a vector of state variables at time t, and $\mathbf{f}(\mathbf{x}(t))$ are the equations defining the dynamics of the system. To determine an optimal model structure and parameterization for the function, $\mathbf{f}$, we begin by collecting time-series data for the system states, $\mathbf{x}(t)$ sampled at times $t_1, t_2, ..., t_m$. This can be arranged in a matrix, $\mathbf{X}$, as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & \mathbf{x_2}(t_m) & \cdots & x_n(t_m) \end{bmatrix}. \tag{2}$$

We then numerically determine the time derivative of these states, $\dot{\mathbf{x}}(t)$, and arrange it in a similar matrix, $\dot{\mathbf{x}}$, as

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}. \tag{3}$$

While we know the model will be composed of a sum of different component functions, we do not know which functions the sparse regression algorithm will select. Therefore, we provide a library of candidate functions, $\Theta(\mathbf{X})$, in the form

$$\Theta(\mathbf{X}) = \begin{bmatrix} \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \cdots & \sin(\mathbf{X}) & e^{\mathbf{X}} & \cdots \end{bmatrix} \tag{4}$$

where $\mathbf{X}^{P_2}$ are possible quadratic nonlinearities in $\mathbf{x}$, $\mathbf{X}^{P_3}$ are possible cubic nonlinearities, and so on. For example,

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & \omega(t_1) & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & \omega(t_2) & x_2^2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \\ x_1^2(t_m) & \omega(t_m) & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix} \tag{5}$$

where $\omega(t) = x_1(t)x_2(t)$ for compactness.
We determine which functions in $\Theta(\mathbf{X})$ will be included in the model by solving the sparse regression problem given by

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi \tag{6}$$

where

$$\Xi = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_n \end{bmatrix} \tag{7}$$

is a matrix of sparse vectors of coefficients. When data is collected from real world experiments or is known to be noisy, an additional $\mathbf{Z}$ matrix can be added to the right side of the sparse regression problem to account for this noise. However, we exclude this term since our simulation data is known to not contain noise, owing to the functioning of the ASPEN Plus Dynamics simulation.

After solving Ξ, the model can be written as

$$\dot{\mathbf{x}}_k = \Theta(\mathbf{x}^T)\boldsymbol{\xi}_k \tag{8}$$

for every row k of the state equations.

To account for some set of input signals, $\mathbf{u}(t)$, driving the system, we assume the system can instead be modeled using state equations of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{9}$$

as shown by [15]. The SINDy algorithm remains unchanged with the regression problem now written as

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}, \mathbf{U})\Xi. \tag{10}$$

3.1.2 SINDy improvement: nonlinear optimization of coefficients

To improve the performance of our SINDy-recovered models, we further optimize the associated sparsity matrix, Ξ, using a constrained nonlinear optimization scheme. This optimization beyond the SINDy method is based on the work of [17], which suggests that while SINDy is capable of finding the location of nonzero elements of Ξ, it cannot necessarily find optimal values for each since Ξ is discontinuous over λ , the sequential least squares thresholding parameter. This parameter

determines the complexity of final model, resulting into sparse models as the parameter is tuned [17]. We apply the method outlined in [17] of sequential quadratic programming (SQP) implemented using MATLAB's *fmincon*. Here we set upper and lower bounds for each nonzero element of Ξ as the given constraints to *fmincon* and construct an optimization function using the mean absolute error (MAE) across all state variables between the training data and the integrated model.

3.1.3 SINDy improvement: inclusion of input derivatives

We consider the time-derivative of each input as an additional input to the system to try to account for hysteresis when modeling streamflow using the SINDy method. For example, the streamflow response will be different for a day of heavy rain following a drought versus following a week of heavy rain, due to water saturation in the soil. We now assume that the system can be represented as a state equation in the form of

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t)), \tag{11}$$

where the sparse regression problem becomes

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}, \mathbf{U}, \dot{\mathbf{U}})\Xi. \tag{12}$$

This additional input ties the current state of the system to past inputs that otherwise could not assert current influence on the system trajectory in the model structure. It should be noted, however, that noise in the measured input signals will propagate to the input derivative and may negatively impact recovered models. Second- and third-order derivatives were tested in the model as well, but discarded when accuracy decreased, possibly due to the propagation of noise mentioned before.

3.2 System selection and data collection

3.2.1 Industrial system: soybean-oil to soybean-diesel plant

For an industrial system case study, we selected a widely used plant system that converts soybean-oil to soybean-diesel. This process uses a series of transesterification reactions in the presence of sodium hydroxide (NaOH) mixed with methanol (MeOH) (Fig. 1). The chemical content of this soybean-oil is provided in [29]. The soybean-oil undergoes a transesterification process in a continuous stirred-tank reactor (CSTR). This reaction produces a mixture of methylated fatty acid molecules, glycerol, unreacted intermediate products, NaOH, and MeOH. The remaining MeOH is then separated from the other components using a RadFrac separation column and reused. A wash column removes the glycerol from the remaining mixture. The soybean-diesel and unreacted intermediate products pass through another RadFrac separation column to separate these two components. The unreacted intermediate products are then mixed with the input stream of soybean-oil to repeat the transesterification process until fully converted. These series produces a complex dynamics due to interaction of several underlying mechanisms. To simplify our model, we fix a number of system parameters to be constant values including:

- MeOH molar flowrate, temperature and pressure;
- NaOH molar flowrate, temperature and pressure;
- Pressure difference of pumps:
- Duty of heat exchangers;
- Vessel geometry of Reactor, MeOH and Diesel RadFrac blocks;
- RadFrac block stage pressures.

State variable selection Selection of state variables to include in a model is a crucial elements of system identification. Failure to select relevant and adequate variables will lead to poor model performance, no matter the tuning of model parameters. In traditional system identification, model structure is typically derived from some understanding of the underlying physics of system. However, with a complex dynamic system, such as the chemical process plant or streamflow system considered here, drawing from first principle understanding may be difficult or impossible due to the sheer number of available measurements or the lack of causal understanding among multiple mechanisms.

Since our primary objective is to achieve a model structure relating the soybean-diesel output to the soybean-oil and water input as molar flow rates, x_1, u_1, and u_2 in Fig. 1, respectively, we select similar molar flow rate state variables with likely relevant dynamics and operating on similar time-scales. We originally chose all molar flowrates between unit operations as the model state variables. However, training and testing of these models quickly reduced to the choice of state variables marked in Fig. 1 as $x_1, \ldots, x_6$, since inclusion of other variables was found to either decrease or at least not increase the accuracy of the model. Additionally,

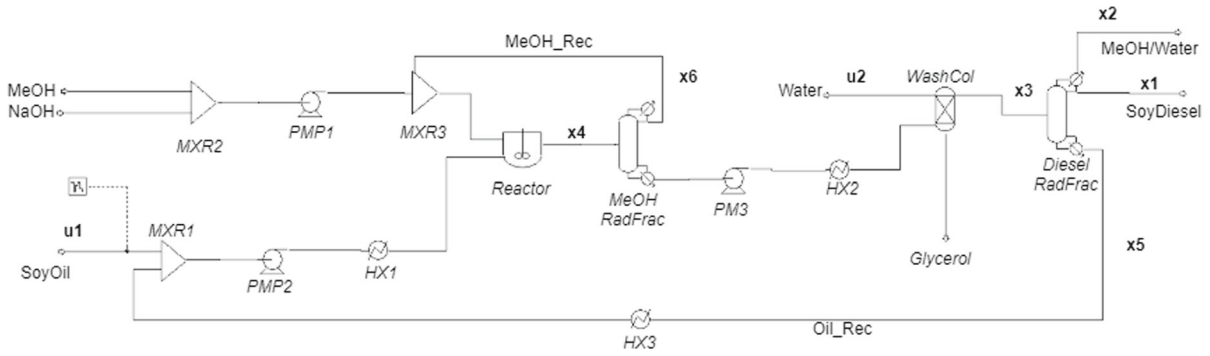

Fig. 1 Soybean-oil to soybean-diesel process with state variables labeled in bold

the glycerol output in Fig. 1 is excluded as preliminary results indicated that the inclusion of this state variable greatly reduced model performance, likely due to nonlinearities in the wash column or dependencies on other variables not considered in the model.

System excitation and data collection Based on [30] we assume that for a system composed of dynamic linearities and static nonlinearities that can be expressed as a block structure, there exists an optimal pseudo-random multilevel sequence generated from Galois field polynomials that is sufficiently exciting. The authors of [30] here define optimal as the minimal number of levels required for excitation. However, they also state that levels beyond this minimum level are not necessarily more exciting to the system. Thus we assume that levels beyond the optimal level are unnecessary but not detrimental to system identification. This last point allows us to make use of the Pseudo-Random Binary Sequence (PRBS) block in ASPEN Plus Dynamics with random amplitude for excitation of the system. By choosing the amplitude to be variable (random), the signal switches from a two-level PRBS to a Galois Field polynomial generated sequence with an arbitrarily high level. Amplitude bounds and period of the signal are then varied until the state variable response appears to oscillate and not simply decrease or increase in the long term. Unfortunately, while this visual observation and adjustment cycle is a successful yet crude strategy for determining an amplitude and frequency that is sufficiently exciting, it fails to detect which of these might drive the system outside of standard operating bounds, or into other dynamic regimes entirely. The system is then simulated for 200 h, and the state variable values are measured every 0.02 h providing 10,000 data points.

3.2.2 Natural system: Lake Vermilion water supply

Lake Vermilion, located in the town of Danville, Illinois, is fed by the North Fork Vermilion River with a watershed of approximately 295 square miles situated in Vermilion and Iroquois Counties, Illinois as well as Warren and Benton Counties, Indiana [31]. The lake was originally formed by damning the North Fork Vermilion River in 1925 and currently holds around three billion gallons of water after the lake level was raised in 1991 due to projected population increase and sedimentation [31]. Sedimentation is estimated to continue at a rate that will reduce the lake water storage capacity by around one-percent per year [32].As of 2008, Lake Vermilion was the municipal water supply for a population of 61,500 spread across the City of Danville, four nearby villages, and much of the surrounding rural area.

State variable selection and data collection To recover a model of water supplied to Lake Vermilion, we use historical climate and streamflow data for the North Fork Vermilion watershed and river, respectively, to train the SINDy algorithm. Climatic factors of solar radiation (R_{solar}), precipitation (P), maximum daily temperature (T_{max}), minimum daily temperature (T_{min}), and vapor pressure deficit (V) between 1950 to 2005 were averaged over the watershed area above the streamflow sampling station located near Bismark, Illinois. This data was obtained from [33]. Streamflow data from the Bismark station is available from November 3, 1988, to September 30, 2010, in 15-min intervals with brief periods (no longer than a week) of missing data [34]. Since the SINDy algorithm requires a numerical time-derivative of state variable data, we use linear interpolation to fill all missing time values. To match the resolution of the climate data, the streamflow data

is summed to daily values. The resulting time period, for which data is available for both the climatic factors as well as streamflow, ranges from 1988 until 2005 resulting in 5589 data points for use in either training or testing at daily resolution.

3.3 Evaluation criteria for selection of models

3.3.1 Evaluation criteria of soybean-diesel plant models

To make an initial determination of model performance that would allow us to select a set of "successful" models, we divide the first 150 h of the process simulation data into five distinct folds, and implement a fivefold cross-validation training scheme. This results in five different models recovered across the training data. We vary the value of the sparsity parameter λ and use this fivefold cross-validation scheme on each value. Each model is integrated over time and the mean absolute error (MAE) is computed using Eq. 13.

$$\text{MAE} = \frac{\sum |x_{r,i} - x_{m,i}|}{n}, \tag{13}$$

In Eq. 13, x_r and x_m are the state variable measurements from the test data and the model estimations, respectively, for each time index i, and n is the size of the test data, between the integrated model and the simulation data from the fold. The model with the lowest MAE of these five models is selected as the representative of this value of λ.

Of these selected models, we make an additional selection based on the lowest MAE value. These high-performing models are then tested on the remaining 50 h of data from the same simulation source as the training data. We also test the model for long term accuracy and stability by testing on 200 h of simulation data-driven by inputs with different random seeds than those in the training, validation, and 50-h test set.

3.3.2 Evaluation criteria of streamflow models

Due to the limited number of data points available for training and testing of the SINDy recovered streamflow models, we use the fivefold validation technique previously described for evaluation of the soybean-diesel plant model as our sole evaluation criteria. Whereas for the soybean-diesel plant we only consider first-order polynomials in the SINDy function library, we expand our search to include second-order polynomials in the streamflow models. Subsequently, we vary the model degree as well as the sparsity parameter and utilize the fivefold cross-validation scheme with MAE (Eq. 13) as the error metric in order to reduce the space of possible models.

4 Results

4.1 Soybean-diesel plant model

4.1.1 Sparsity adjustment for SINDy and model training

We explore a range of thresholding values to vary the sparsity of the coefficient matrix for models with a function library including only first-order functions. As a metric of comparison between models, we use MAE averaged over all five validation folds for each model. We first explore the model performance for $0 \leq \lambda \leq 0.1$ with a resolution of 0.0025. Values of λ around 0.1 result in an overly sparse matrix with all terms equal to zero, while $\lambda = 0$ results in no forced sparsity. The resulting plot of mean error between all six state variables as well as for x_1 individually is shown in Fig. 2 for both the SINDy derived models as well as those further optimized. We see in Fig. 2 that the optimized models do not necessarily result in lower error versus the standard SINDy models when compared against the validation data.

In Fig. 2, we see minimized error between all six state variables at $\lambda = 0.025$, and minimized error for x_1 at $\lambda = 0.01$ and $\lambda = 0.08$. The ranges of λ around these values are further explored with a resolution of 0.001 to verify that error cannot be reduced further, however, adjacent values of λ with higher performing models were not found.

4.1.2 Fivefold cross-validation

The MAE for each of the five validation folds is shown in Table 1 for the three models with sparsity parameters of $\lambda = 0.01, 0.025, 0.08$. We look at the model results for the average of the six state variables as well as x_1 specifically, since accurate modeling of the output is a priority. Looking at the MAE values for each fold, we see that the model with the lowest error for both x_1 (soybean-diesel out) as well as the average of all

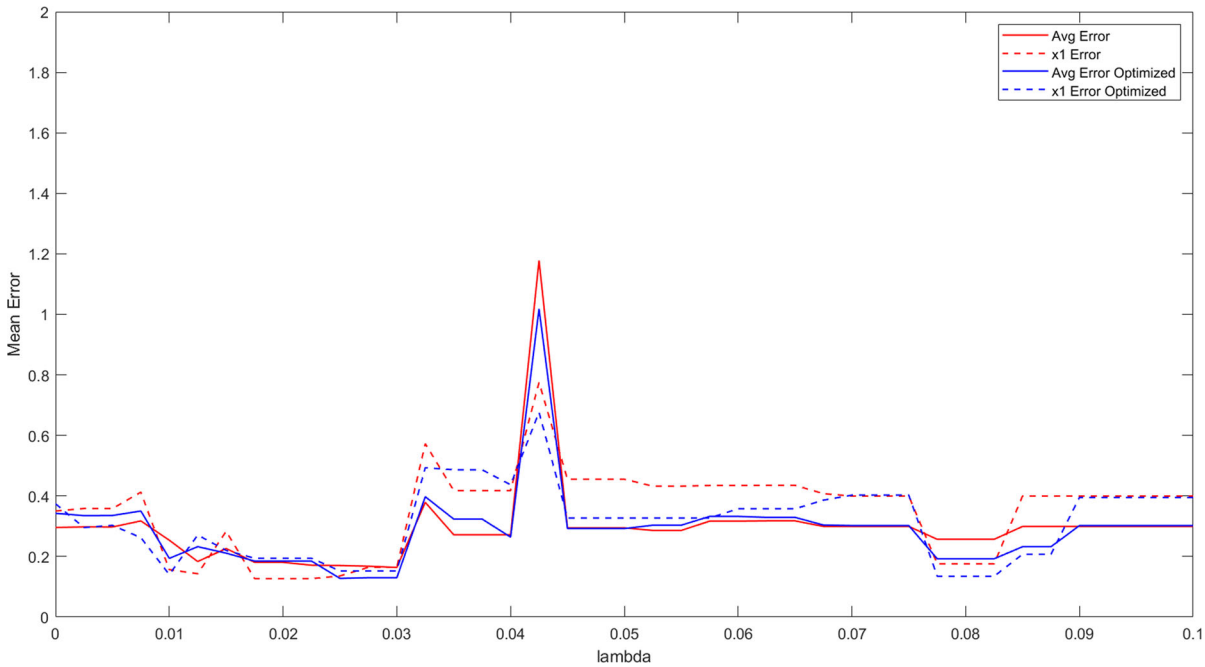

Fig. 2 Minimum mean error over each set of five validation folds for $0 \leq \lambda \leq 0.1$

variables varies between the standard SINDy derived model and the optimized model. This matches what can be seen in Fig. 2 across all models, which suggests that the added optimization is not necessarily resulting in overfitting to the training data. Additionally, we see that validating on the first fold tends to yield the highest error, which indicates there may be certain dynamics during the initialization of the system that do not continue in normal operation. For the models with $\lambda = 0.01, 0.025$, the lowest error is found in the third validation fold, while for the model with $\lambda = 0.08$, the lowest error is found in the fifth validation fold. Lowest error in the third validation fold is expected due to the symmetry of training data and the placement of the third validation fold in the middle. Overall, the lowest error is found in the third validation fold of the model with $\lambda = 0.025$.

4.1.3 Model testing

We use the same error metric of MAE to judge the performance of the the models with $\lambda = 0.01, 0.025, 0.08$ on the test datasets of 50 h and 200 h. These results are tabulated in Table 2.

We see that for the dataset comprised of the 50 h of simulation data following the training data, the standard SINDy model outperforms the optimized model for both the average error between all six state variables as well as the x_1 individually. However, for the long term test data of 200 h, the optimized model performs significantly better. For the 50 h of test data, the standard SINDy model with $\lambda = 0.01$ results in the lowest error. However, for the long term dataset of 200 h, the optimized model with $\lambda = 0.025$ results in the lowest error, while the standard SINDy model with $\lambda = 0.01$ results in the highest amount of error. This seems to suggest that the standard SINDy models may be overfitting to a particular aspect of the simulation data from which both the training dataset and 50 h dataset are taken.

Both the standard SINDy derived model with $\lambda = 0.025$ and the further optimized versions are integrated over time using the 50 h and 200 h test dataset inputs. The resulting plots can be seen in Figs. 3 and 4, respectively. In Fig. 3, we see that both the standard and optimized models fit x_4 and x_6 very well. For x_1, x_2, and x_3 in Fig. 3, the model captures the lower frequency oscillations, but fails to reconstruct higher frequency changes. The model recreation of these state variable dynamics over time also appears to be slowly diverging from the test data, likely due to accumulation of error in the numerical integration. The worst model performance is clearly seen in the model reconstruction of x_5.

Table 1 Error as MAE between five validation folds and integrated models, where x_1 is the molar flow rate of soy-diesel out, x_2 is MeOH/water mixture out, x_3 is the flow rate between the WashCol and diesel RadFrac blocks, x_4 is the flow rate between the reactor and MeOH RadFrac blocks, x_5 is the oil recycling loop from the diesel RadFrac column bottom, and x_6 is the recycling loop from the top of the MeOH RadFrac block (see Fig. 1)

Variable(s)	Optimized	λ	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
avg(x_1, ..., x_6)	No	0.01	1.0980	0.3590	0.2537	0.6823	0.3175
x_1	No	0.01	0.9920	0.4983	0.1565	0.8518	0.4125
avg(x_1, ..., x_6)	Yes	0.01	0.6116	1.1461	0.1939	0.4164	0.3649
x_1	Yes	0.01	0.4408	1.8151	0.1404	0.3614	0.5263
avg(x_1, ..., x_6)	No	0.025	1.4653	0.9635	0.1690	0.8549	0.1716
x_1	No	0.025	1.7164	1.4017	0.1376	1.1080	0.2227
avg(x_1, ..., x_6)	Yes	0.025	0.9535	1.9236	0.1280	0.7845	0.2754
x_1	Yes	0.025	0.9513	2.8761	0.1479	0.9826	0.3526
avg(x_1, ..., x_6)	No	0.08	1.7184	0.5653	0.4265	0.5408	0.2991
x_1	No	0.08	2.3015	0.9276	0.7369	0.5225	0.3998
avg(x_1, ..., x_6)	Yes	0.08	1.2498	0.3070	0.4016	0.5083	0.3022
x_1	Yes	0.08	1.7074	0.4128	0.6963	0.3952	0.4149

Table 2 Error as MAE between both short and long sets of test data and integrated models, where x_1 is the molar flow rate of soy-diesel out, x_2 is MeOH/Water mixture out, x_3 is the flow rate between the WashCol and Diesel RadFrac blocks, x_4 is the flow rate between the Reactor and MeOH RadFrac blocks, x_5 is the oil recycling loop from the Diesel RadFrac column bottom, and x_6 is the recycling loop from the top of the MeOH RadFrac block (see Fig. 1)

Variable(s)	Optimized	Test data	$\lambda = 0.01$	$\lambda = 0.025$	$\lambda = 0.08$
avg(x_1, ..., x_6)	No	50 h	0.1521	0.4734	0.2678
x_1	No	50 h	0.1672	0.6620	0.3921
avg(x_1, ..., x_6)	Yes	50 h	0.6038	0.5370	0.2945
x_1	Yes	50 h	0.8847	0.7570	0.4422
avg(x_1, ..., x_6)	No	200 h	5.3239	2.1925	1.3440
x_1	No	200 h	8.5559	3.3656	0.7230
avg(x_1, ..., x_6)	Yes	200 h	0.5255	0.2522	1.2656
x_1	Yes	200 h	0.7524	0.2948	0.5924

Interestingly, across all state variables the reconstruction of the standard SINDy model is closer to the 50 h test data than the reconstruction using the optimized model, which further suggests some type of overfitting to a particular aspect of this particular set of simulation data.

In Fig. 4, we see that in the case of all variables but x_4 and x_6, the standard SINDy model quickly diverges from the 200 h test data. However, unlike in Fig. 3, this divergent behavior is not seen in the optimized model, which while still failing to capture all high frequency oscillations in x_1, x_2, and x_3, is a much closer reconstruction. Additionally, the reconstruction of x_5 using the optimized model in Fig. 4 is accurate.

4.1.4 Soybean-diesel model structure

To compare the structure of terms between models, we use the optimized model from each of the lowest error validation folds in Table 1. Since the optimization is only applied to nonzero terms and is bounded above and below, the structure of these state equations is identical to the standard SINDy recovered equations with only some changes to parameter values. For $\lambda = 0.01$

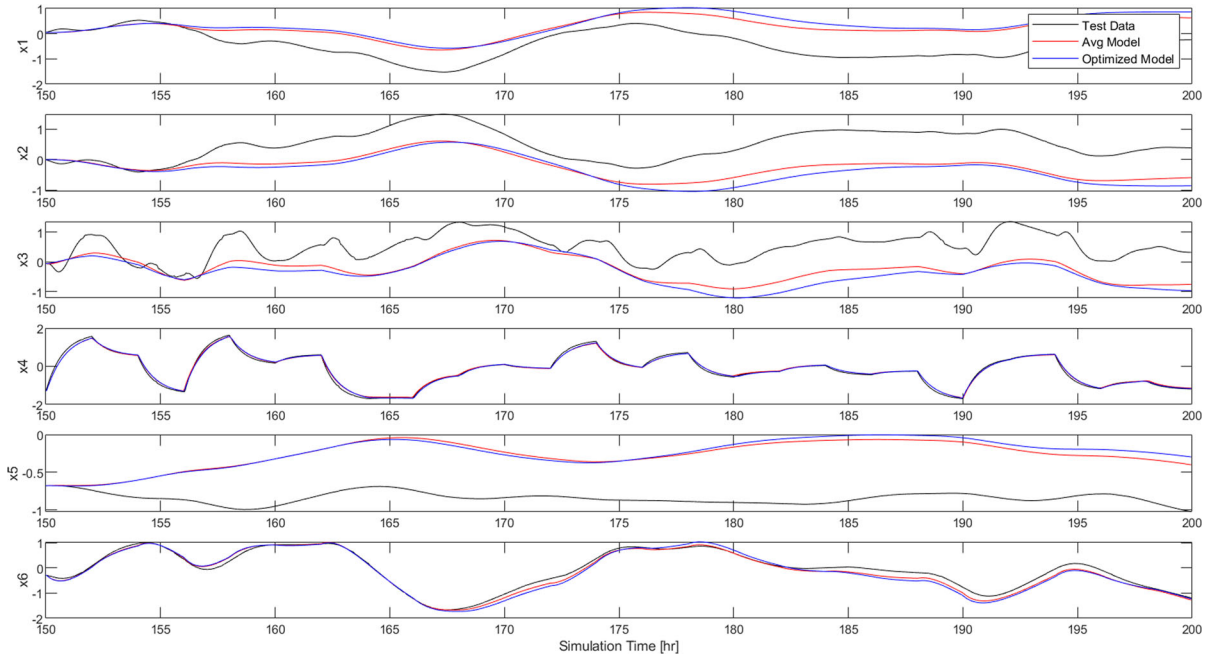

Fig. 3 Integrated model results over 50 h of test data for$\lambda = 0.025$

we have the model from the third validation fold of

$$\begin{aligned}
\dot{x}_1 &= 0.0243\,u_2 - 1.31\,x_1 - 1.72\,x_2 + 0.451\,x_3 \\
&\quad - 0.0344\,x_4 - 0.0696\,x_5 - 0.0289\,x_6 \\
\dot{x}_2 &= 1.27\,x_1 - 0.0273\,u_2 - 0.00935\,u_1 + 1.66\,x_2 \\
&\quad - 0.431\,x_3 + 0.0215\,x_4 + 0.0735\,x_5 + 0.0282\,x_6 \\
\dot{x}_3 &= 1.09\,x_1 - 0.0332\,u_2 - 0.0738\,u_1 + 1.72\,x_2 \\
&\quad - 0.605\,x_3 + 0.151\,x_4 + 0.0844\,x_5 \\
&\quad - 0.123\,x_6 + 0.027 \\
\dot{x}_4 &= 0.0117\,u_2 - 1.73\,u_1 + 0.0626\,x_2 - 0.0823\,x_3 \\
&\quad - 1.3\,x_4 + 0.0192\,x_6 + 0.00423 \\
\dot{x}_5 &= 0.0385\,x_6 - 0.534\,x_2 - 0.00384\,x_3 - 0.0693\,x_5 \\
&\quad - 0.544\,x_1 - 0.0233 \\
\dot{x}_6 &= 0.174\,u_1 - 0.0055\,u_2 - 0.607\,x_1 - 0.945\,x_2 \\
&\quad + 0.354\,x_3 + 0.472\,x_4 - 0.0775\,x_5 - 0.135\,x_6
\end{aligned} \tag{14}$$

for $\lambda = 0.025$ the third validation fold model is given by

$$\begin{aligned}
\dot{x}_1 &= 0.0317\,u_2 - 1.16\,x_1 - 1.57\,x_2 + 0.436\,x_3 \\
&\quad - 0.046\,x_4 - 0.0685\,x_5 - 0.0218\,x_6 \\
\dot{x}_2 &= 1.12\,x_1 - 0.0331\,u_2 + 1.52\,x_2 - 0.424\,x_3 \\
&\quad + 0.0455\,x_4 + 0.0702\,x_5 + 0.0171\,x_6 \\
\dot{x}_3 &= 0.885\,x_1 - 0.0886\,u_1 + 1.56\,x_2 - 0.659\,x_3 \\
&\quad + 0.138\,x_4 + 0.0737\,x_5 - 0.124\,x_6 \\
\dot{x}_4 &= 0.138\,x_2 - 1.63\,u_1 - 0.142\,x_3 - 1.18\,x_4 \\
\dot{x}_5 &= 0.0367\,x_6 - 0.441\,x_2 - 0.0457\,x_5 - 0.454\,x_1 \\
\dot{x}_6 &= 0.181\,u_1 - 0.548\,x_1 - 0.893\,x_2 + 0.36\,x_3 \\
&\quad + 0.469\,x_4 - 0.0739\,x_5 - 0.139\,x_6
\end{aligned} \tag{15}$$

and for $\lambda = 0.08$ we have the fifth validation fold model in the form of

$$\begin{aligned}
\dot{x}_1 &= 0.373\,x_3 - 0.729\,x_2 - 0.399\,x_1 \\
\dot{x}_2 &= 0.328\,x_1 + 0.634\,x_2 - 0.35\,x_3 \\
\dot{x}_3 &= 0.993\,x_1 + 1.77\,x_2 - 0.751\,x_3 + 0.245\,x_4 \\
&\quad + 0.0816\,x_5 - 0.147\,x_6 \\
\dot{x}_4 &= -1.8\,u_1 - 1.35\,x_4 \\
\dot{x}_5 &= 0 \\
\dot{x}_6 &= 0.168\,u_1 - 0.65\,x_1 - 1.02\,x_2 + 0.381\,x_3 \\
&\quad + 0.456\,x_4 - 0.086\,x_5 - 0.13\,x_6
\end{aligned} \tag{16}$$

We see that as the sparsity parameter λ increases, the sparsity of the model increases as expected with terms dropping out between $\lambda = 0.01$ to $\lambda = 0.08$. Several of these terms include the water input u_2, which falls out of $\dot{x}_4$ and $\dot{x}_6$ between Eqs. 14 and 15. With $\lambda = 0.08$,

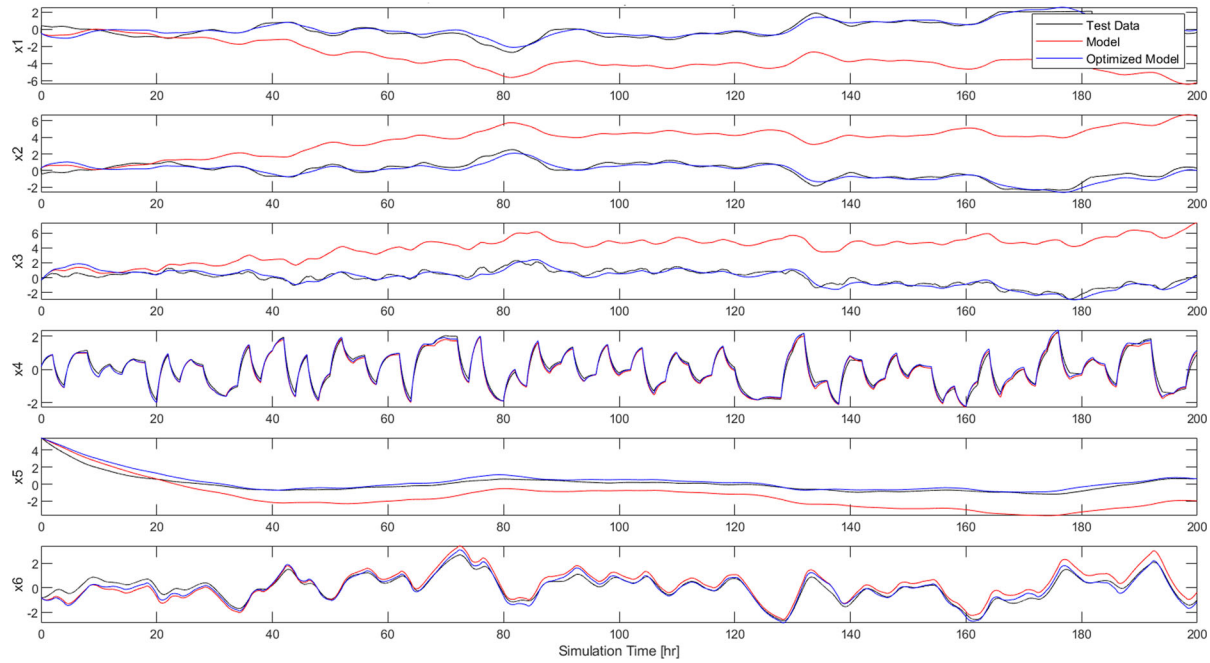

Fig. 4 Integrated model results over 200 h of test data for $\lambda = 0.025$.

terms including u_2 in any of the state equations are gone. Additionally, with $\lambda = 0.08$, the equation for $\dot{x}_5$ is overly sparse with only a constant zero term. Ultimately, as the sparsity is increased all state equations will go to zero.

As terms fall out due to the increased sparsity, some remaining terms retain coefficients in the same vicinity as the previous equations, while others change significantly. For example, $\dot{x}_4$ loses all terms except for u_1 and x_4, but keeps similar coefficients, while $\dot{x}_1$ loses all terms except for x_1, x_2, and x_3, of which x_1 and x_2 are significantly different.

From these equations, we can see that a basic mass balance is captured along with other expected linear behavior. In Eq. 15, we see that the rate of change of x_1, the soybean diesel output, is impacted positively by x_3, the input to the Diesel RadFrac column in Fig. 1, and u_2, the water input driving the separation process in the WashCol in Fig. 1. Likewise, this rate of change is negatively impacted by x_2 and x_5, both additional outputs from the Diesel RadFrac distillation column. The recycling stream from the MeOH RadFrac distillation column, x_6, also has a negative impact on the rate of change in x_1. Perhaps surprisingly, x_1 has a large negative impact on its own rate of change. This is likely due to the dynamics of saturation in the distillation column (i.e., a larger value of x_1 means that in the next time step the Diesel RadFrac column produces lesser output possibly due to hold up in the column). The structure of the state equation for x_2 inversely mirrors that of x_1. For the rate of change of x_3, the non-Glycerol output of the WashCol in Fig. 1, we see a positive impact from x_1, x_2, x_5, and x_4, and a negative impact from x_6, which makes intuitive sense from a mass balance view. Additionally, we see a negative impact from the soybean oil input, u_1, and x_3 itself likely due to saturation dynamics. In the state equation for x_4, the reactor output, we see a positive impact from x_2 and large negative impacts from the soybean oil input, u_1, as well as x_3 and x_4 also likely capturing the dynamics of over saturation. For the state equation for x_5, we see a negative impact from x_1 and x_2, reflecting the mass balance at the Diesel RadFrac column, and a small negative impact from x_5 itself, likely capturing the effect of over saturation. Lastly, for the recycled flow stream, x_6, emerging from the MeOH RadFrac column, we see a strong positive impact from x_4 and u_1, which makes sense from a mass balance perspective. Interestingly, there is an additional positive impact from x_3, while x_1, x_2, and x_5 all provide a negative impact on the rate of change of x_6, indicating a reduction in mass being recycled as the outputs are increased. These terms and approximate parame-

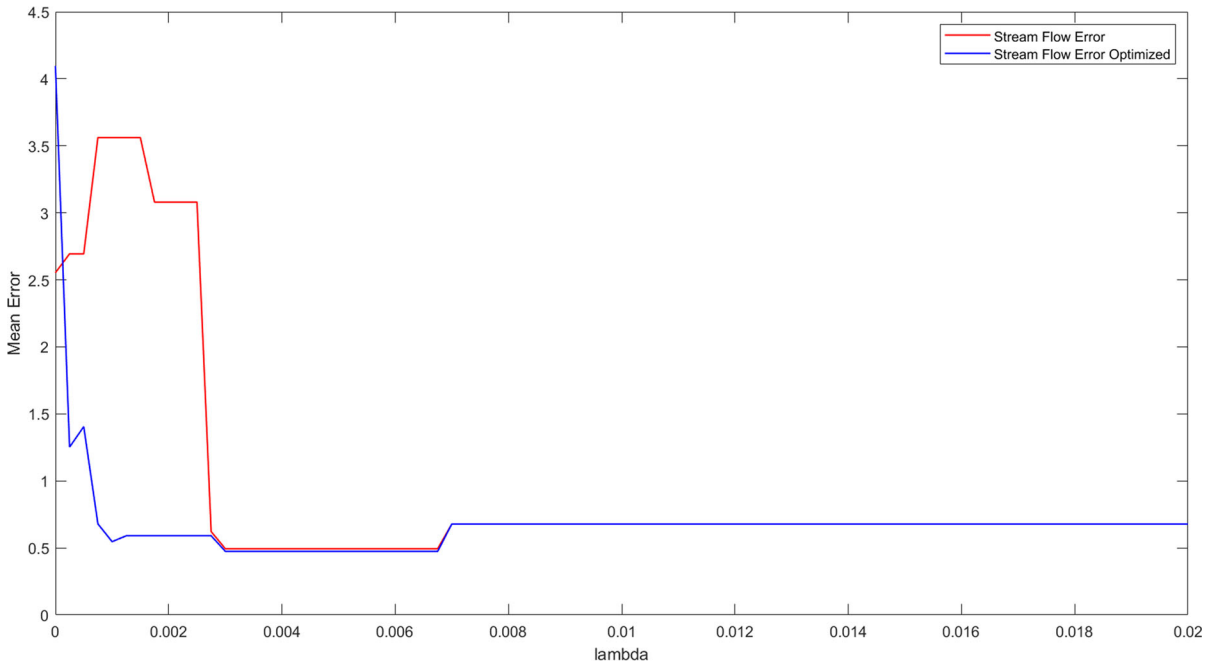

Fig. 5 Minimum mean error over each set of five validation folds for $0 \leq \lambda \leq 0.02$

ter values persist even when the sparsity is somewhat increased, as evidenced in Eq. 16.

4.2 Stream flow model

4.2.1 Sparsity adjustment for SINDy and model training

We again explore the range of thresholding values for which the model sparsity will change. However, unlike the soybean-diesel plant models discussed previously, we also consider second-order polynomials in the function library supplied to the SINDy algorithm. Additionally, we consider models for which the input derivatives are included as inputs themselves. The accuracy metric of MAE is averaged over all five validation folds for each standard SINDy and further optimized model. We explore model performance over $0 \leq \lambda \leq 0.02$ for both the linear and nonlinear function libraries as seen in Figs. 5, 6, and 7 for the first-order polynomial function library, second-order polynomial function library, and second-order polynomial library with inclusion of input derivatives, respectively.

From Fig. 5 we choose to further examine $\lambda = 0.006$, as there is no change in performance in the surrounding space. From Fig. 6, we see lowest error values for λ at 0.0005 and 0.00125; however, we also choose to further examine $\lambda = 0.012$ as a model in the last range of λ before model reduction to 0. Lastly, from Fig. 7, we choose to further examine models for which $\lambda = 0.00175$ and $\lambda = 0.00325$. As seen in Figs. 6 and 7, the further optimized models no longer result in reduced MAE for all values of λ, despite providing a better approximation of abrupt changes in streamflow than that of the standard model.

4.2.2 Fivefold cross-validation and model testing

The MAE for each of the five validation folds is shown in Table 3, where the sparsity parameter of $\lambda = 0.006$ is considered as a linear model and $\lambda = 0.0005, 0.00125, 0.012$ are considered for nonlinear models. Models that failed during integration, due either to stiffness or unbounded behavior, are marked with 'NaN' rather than an MAE error value.

In Figs. 8 and 9 we plot the results of the integrated optimized nonlinear models for $\lambda = 0.0005$ and $\lambda = 0.00125$, respectively. We see that both models appear to capture seasonal streamflow tendencies, yet fail to respond to more abrupt changes happening over weeks or months.

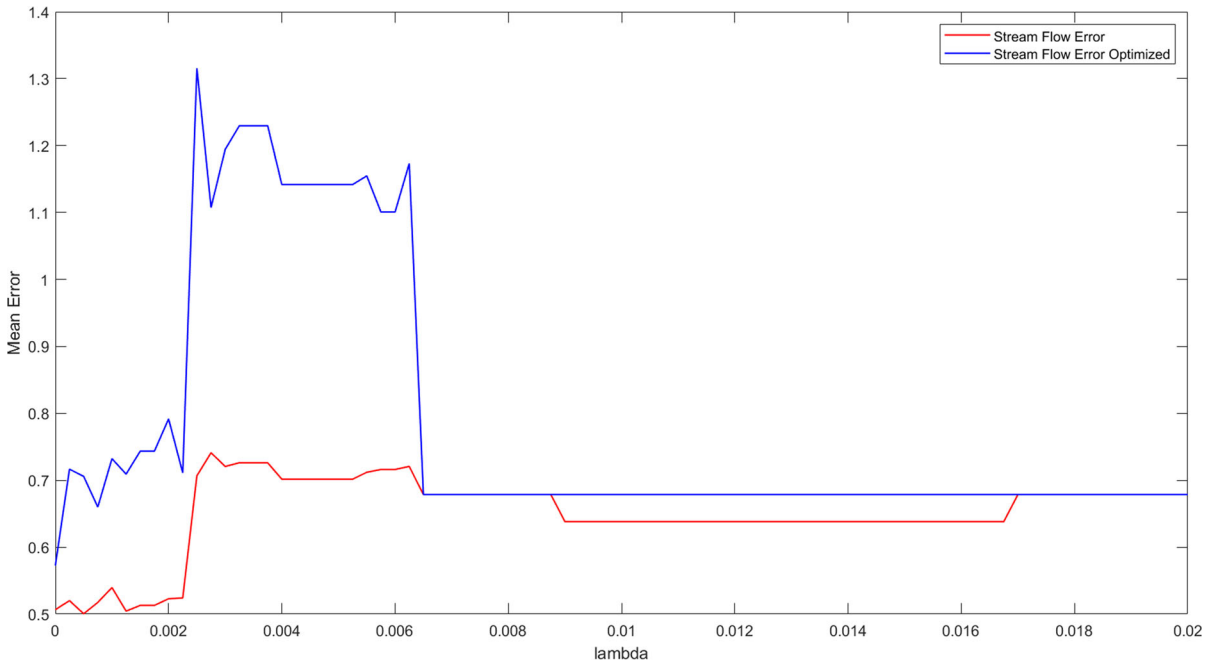

Fig. 6 Minimum mean error over each set of five validation folds for $0 \leq \lambda \leq 0.02$

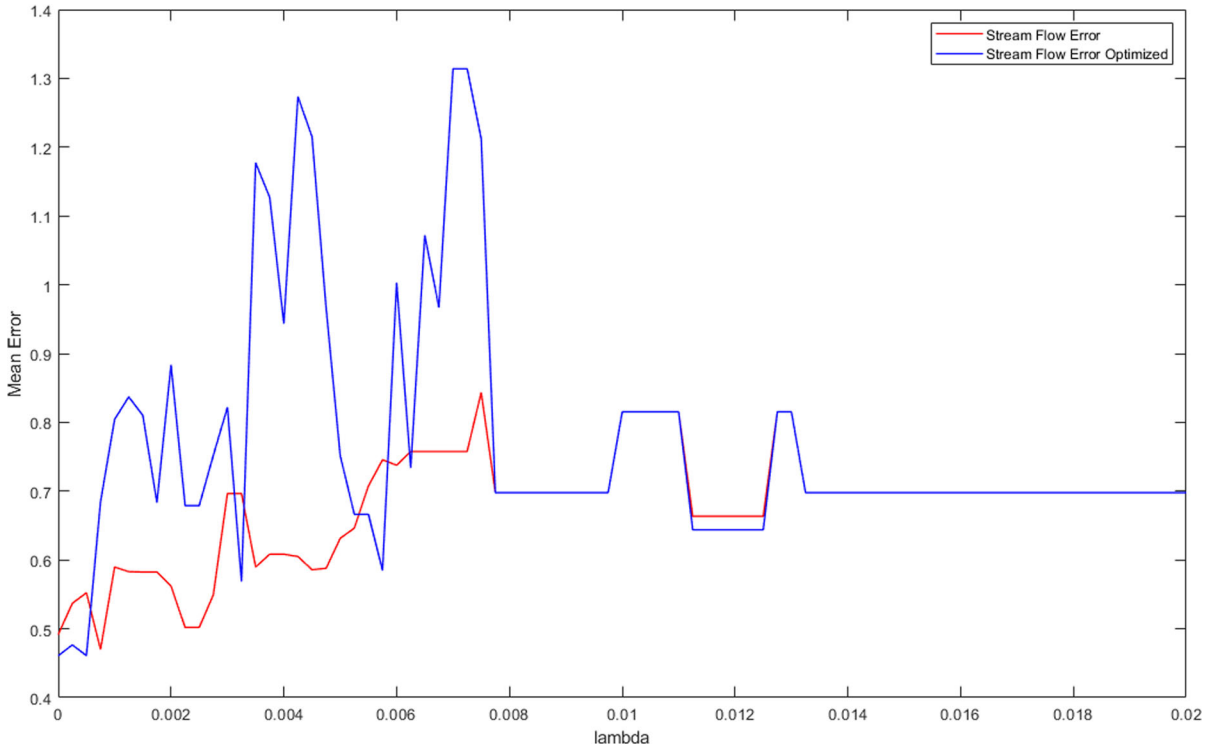

Fig. 7 Minimum mean error over each set of five validation folds for $0 \leq \lambda \leq 0.02$

Table 3 Error as MAE between five validation folds and integrated models

Optimized	Order	$\dot{u}$	λ	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
No	1	No	0.006	0.6978	0.9285	0.6334	1.0732	0.8815
Yes	1	No	0.006	0.6978	1.1298	0.6416	0.8998	1.2655
No	2	No	0.0005	0.5171	NaN	0.7712	NaN	1.1387
Yes	2	No	0.0005	0.4701	NaN	0.8118	NaN	1.2165
No	2	No	0.00125	1.0376	1.2995	0.9083	1.2114	0.9594
Yes	2	No	0.00125	0.6302	5.1275	0.7418	16.9459	1.2391
No	2	No	0.012	0.6978	8.2418	9.8995	1.4322	6.0414
Yes	2	No	0.012	0.6978	0.9560	0.7268	1.4322	1.3760
No	2	Yes	0.00175	0.5825	0.7522	0.6061	6.8310	1.0880
Yes	2	Yes	0.00175	0.9897	3.1199	0.6834	27.9328	NaN
No	2	Yes	0.00325	0.8024	0.7121	0.6966	0.8955	1.1799
Yes	2	Yes	0.00325	0.5688	4.8437	0.8218	1.3573	1.5917

By contrast, in Figs. 10 and 11 of integrated models containing input derivative terms, we see much better model reconstruction of individual streamflow peaks within a season of increased streamflow. However, these models still fail to reach the upper ranges of streamflow values and also fail to remain at low values for periods where streamflow is minimal.

4.2.3 Streamflow model structure

We compare the structure of terms between the optimized nonlinear models for $\lambda = 0.00125$, the standard SINDy model, and $\lambda = 0.00175$, the model containing input derivative terms. These equations are given by Eqs. 17 and 18, respectively, where Q_{stream}, R_{solar}, P, T_{max}, T_{min}, and V refer to the river flow rate, solar radiation, precipitation, maximum daily temperature, minimum daily temperature, and vapor pressure deficit.

$$\begin{aligned}\dot{Q}_{\text{stream}} = {} & 0.00781\,P - 0.024\,T_{\text{min}} + 0.0191\,T_{\text{max}} \\ & - 0.0112\,V - 0.0138\,P\,Q_{\text{stream}} \\ & - 0.0048\,P\,R_{\text{solar}} - 0.00284\,Q_{\text{stream}}\,R_{\text{solar}} \\ & + 0.00352\,P\,T_{\text{min}} - 0.0144\,P\,T_{\text{max}} \\ & + 0.0276\,Q_{\text{stream}}\,T_{\text{min}} - 0.0248\,Q_{\text{stream}}\,T_{\text{max}} \\ & + 0.0157\,P\,V + 0.009\,R_{\text{solar}}\,T_{\text{min}} \\ & + 0.00673\,R_{\text{solar}}\,T_{\text{max}} - 2.71e-4\,R_{\text{solar}}\,V \\ & + 0.00329\,T_{\text{min}}\,T_{\text{max}}\,T_{\text{max}}\,V - 0.0159\,{T_{\text{min}}}^2 \\ & - 0.00754\,{T_{\text{max}}}^2 - 0.00347\,V^2 - 7.85e-4 \end{aligned} \tag{17}$$

We see terms present in Eq. 17 repeated again in Eq. 18 with different magnitudes but the same positive or negative impact on $\dot{Q}_{\text{stream}}$. For example the term $-0.00754T_{\text{max}}^2$ present in Eq. 17 is also present as $-0.0158T_{\text{max}}^2$ in Eq. 18. The fact that these terms remain even with consideration of additional inputs, suggests that, while unknown, they do have some relevant physical interpretation and are not the result of overfitting to noise in the streamflow or climate data.

In Eq. 18 we see that for many terms the positive or negative impact is switched when one of the component input variables is switched with its derivative. For example, the positive $R_{\text{solar}}V$ term becomes negative when either $\dot{R}_{\text{solar}}$ or $\dot{V}$ is substituted for R_{solar} or V, respectively, but remains positive when both are substituted. This makes sense if we consider the effect of past system inputs on current system state. A positive value for either $\dot{R}_{\text{solar}}$ or $\dot{V}$ implies that the current value is larger than the previous input value and this previously smaller value exerts a negative impact on the rate of change of streamflow. Conversely, a negative value for either $\dot{R}_{\text{solar}}$ or $\dot{V}$ implies that the current value is smaller and this past large value exerts a positive impact on $\dot{Q}_{\text{stream}}$. The num-

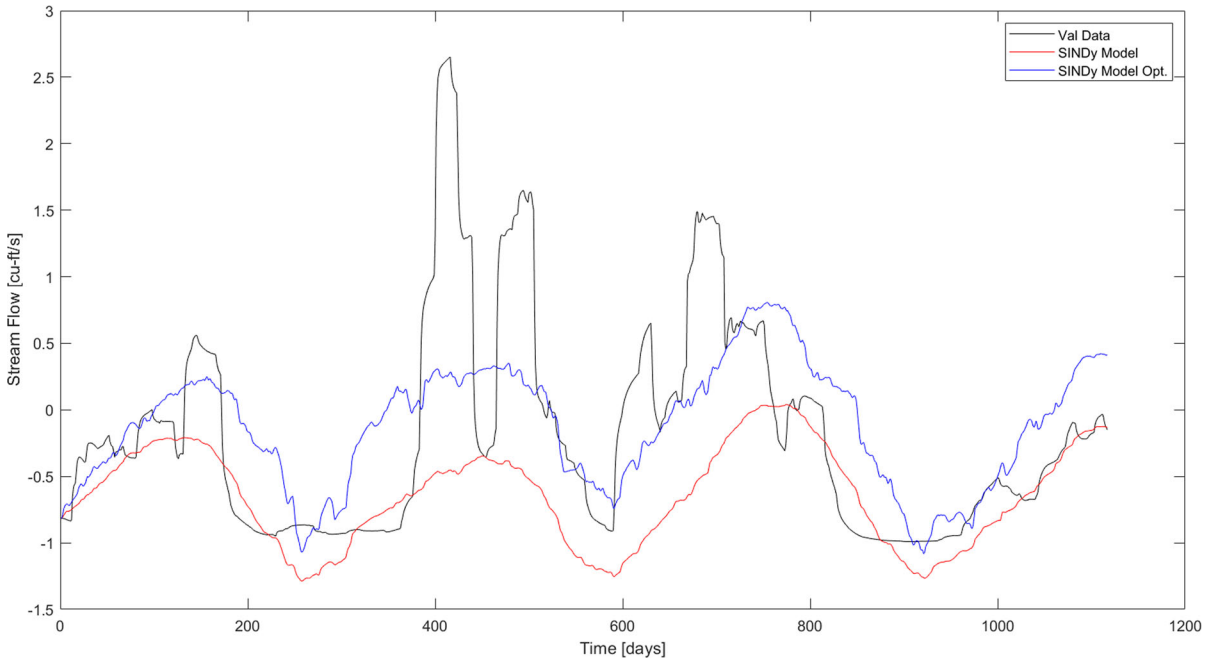

Fig. 8 Nonlinear integrated model results over first validation fold for $\lambda = 0.0005$

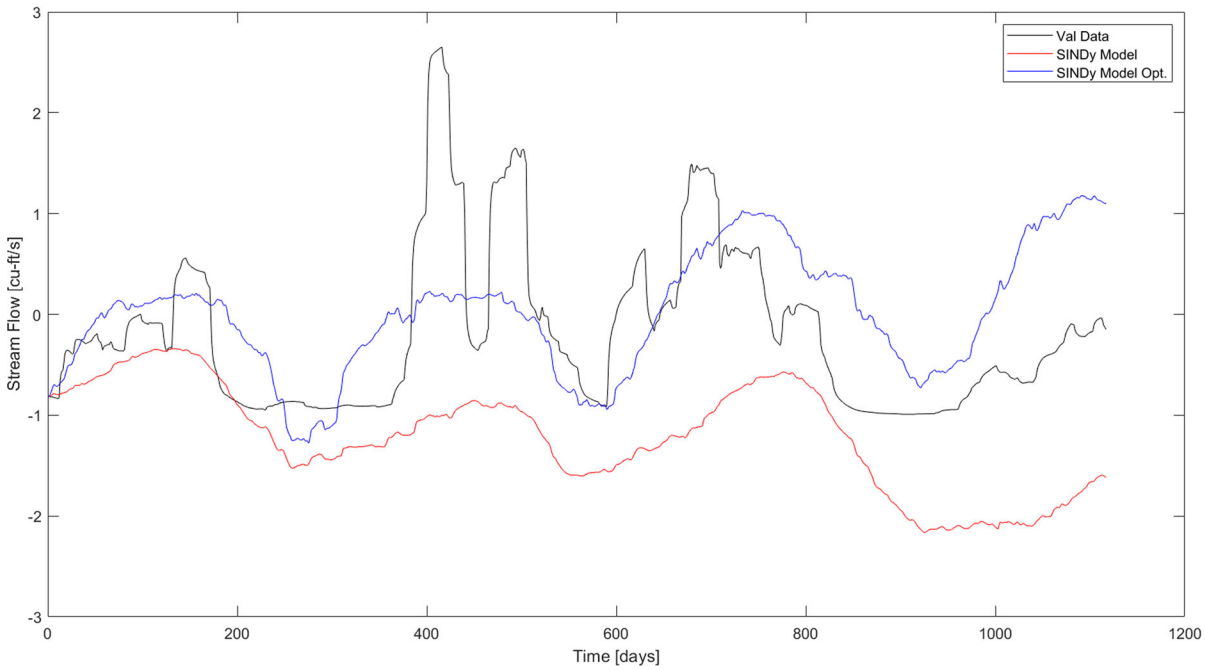

Fig. 9 Nonlinear integrated model results over first validation fold for $\lambda = 0.00125$

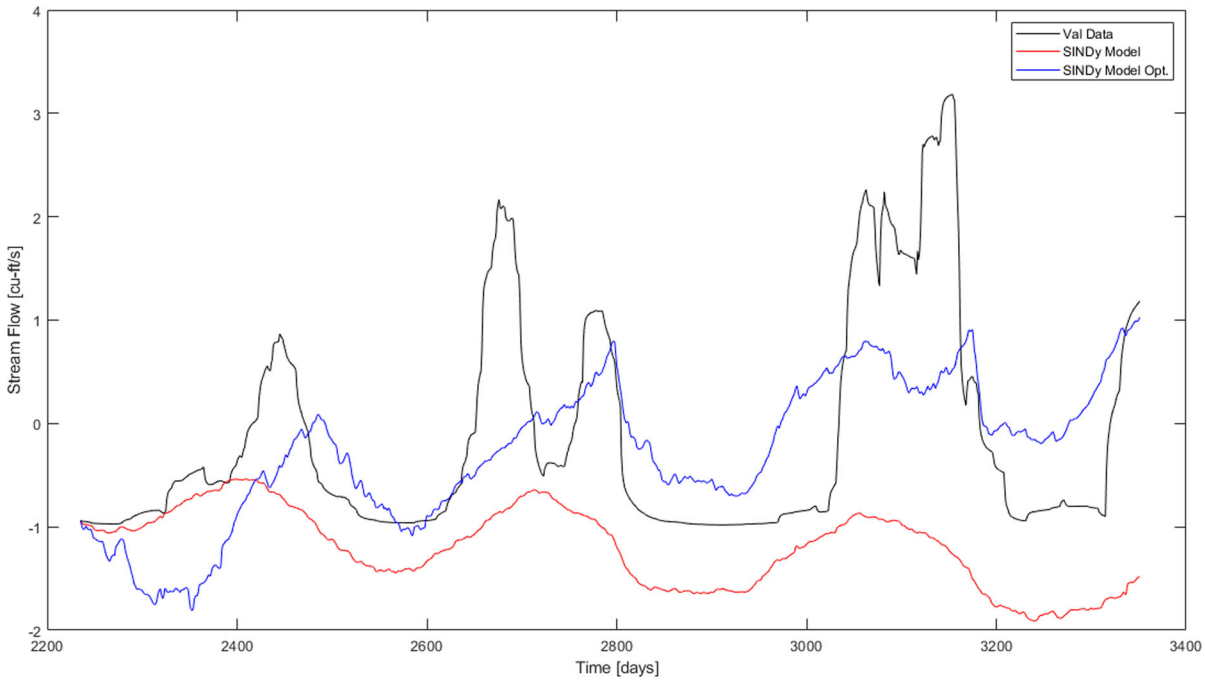

Fig. 10 Nonlinear with input derivative integrated model results over first validation fold for $\lambda = 0.00175$

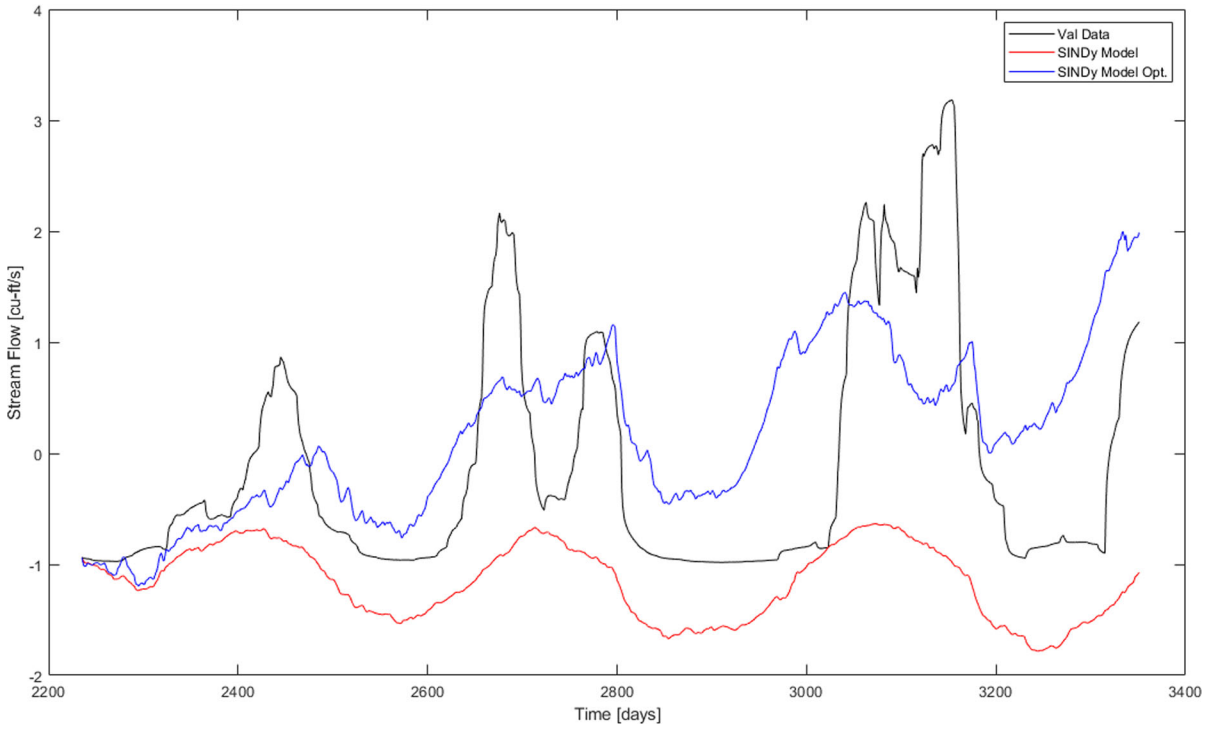

Fig. 11 Nonlinear with input derivative integrated model results over first validation fold for $\lambda = 0.00325$

ber of input derivative terms appearing in the summation suggests a heavy dependence on input history and additional steps to address this hysteresis are likely necessary to improve the accuracy of the model.

$$
\begin{aligned}
\dot{Q}_{\text{stream}} = {} & 0.00619\, P - 0.00656\, R_{\text{solar}} + 0.0167\, \dot{R}_{\text{solar}} \\
& -0.0475\, T_{\text{min}} + 0.0707\, T_{\text{max}} + 0.0121\, \dot{T}_{\text{min}} \\
& -0.0141\, \dot{T}_{\text{max}} - 0.0208\, V - 0.00707\, \dot{V} \\
& -0.0235\, P\, Q_{\text{stream}} + 0.0133\, \dot{P}\, Q_{\text{stream}} \\
& -0.00506\, Q_{\text{stream}}\, R_{\text{solar}} + 0.00593\, Q_{\text{stream}}\, \dot{R}_{\text{solar}} \\
& +0.0334\, P\, T_{\text{min}} - 0.0497\, P\, T_{\text{max}} - 0.0156\, P\, \dot{T}_{\text{min}} \\
& -0.0343\, P\, \dot{T}_{\text{max}} - 0.0173\, \dot{P}\, T_{\text{min}} \\
& +0.0314\, \dot{P}\, T_{\text{max}} + 0.0129\, \dot{P}\, \dot{T}_{\text{min}} \\
& +0.0224\, \dot{P}\, \dot{T}_{\text{max}} + 0.00748\, Q_{\text{stream}}\, T_{\text{min}} \\
& -0.00738\, Q_{\text{stream}}\, T_{\text{max}} + 0.0128\, Q_{\text{stream}}\, \dot{T}_{\text{max}} \\
& +0.0304\, P\, V + 0.00566\, P\, \dot{V} - 0.0201\, \dot{P}\, V \\
& -0.0112\, R_{\text{solar}}\, T_{\text{min}} + 0.0282\, R_{\text{solar}}\, T_{\text{max}} \\
& +0.0253\, \dot{R}_{\text{solar}}\, T_{\text{min}} - 0.0233\, R_{\text{solar}}\, \dot{T}_{\text{max}} \\
& -0.0198\, \dot{R}_{\text{solar}}\, T_{\text{max}} - 0.0266\, \dot{R}_{\text{solar}}\, \dot{T}_{\text{max}} \\
& +0.00654\, Q_{\text{stream}}\, V + 0.00344\, Q_{\text{stream}}\, \dot{V} \\
& +0.00506\, R_{\text{solar}}\, V - 0.0218\, \dot{R}_{\text{solar}}\, V - 0.0171\, R_{\text{solar}}\, \dot{V} \\
& +0.0384\, \dot{R}_{\text{solar}}\, \dot{V} + 0.0211\, T_{\text{min}}\, T_{\text{max}} + 0.0572\, T_{\text{min}}\, \dot{T}_{\text{min}} \\
& -0.0239\, T_{\text{min}}\, \dot{T}_{\text{max}} - 0.0462\, T_{\text{max}}\, \dot{T}_{\text{min}} \\
& +0.0579\, T_{\text{max}}\, \dot{T}_{\text{max}} - 0.00872\, \dot{T}_{\text{min}}\, \dot{T}_{\text{max}} \\
& +0.0099\, T_{\text{min}}\, V + 0.0289\, T_{\text{max}}\, V - 0.0646\, \dot{T}_{\text{max}}\, V \\
& +0.0163\, T_{\text{min}}\, \dot{V} - 0.0112\, T_{\text{max}}\, \dot{V} + 0.0623\, \dot{T}_{\text{max}}\, \dot{V} \\
& +0.0189\, V\, \dot{V} - 0.00742\, {R_{\text{solar}}}^2 \\
& -0.0195\, {T_{\text{min}}}^2 - 0.0158\, {T_{\text{max}}}^2 - 0.0374\, \dot{T}_{\text{max}}^2 \\
& -0.0113\, V^2 - 0.0206\, \dot{V}^2 - 0.0112
\end{aligned}
\tag{18}
$$

5 Conclusion and discussion

In this paper, we propose using a recent system identification method based on sparse regression and optimization of model coefficients for fast recovery of low-order dynamic models of process industries and natural systems. For the industrial system, we utilize a hybrid mechanistic-machine learning approach by using the simulated data for process flow obtained from high-order mechanistic models and identify a low-order model using machine learning. Similarly, for the natural system we use data from observations and complex climate models based on physical principles. We modify the original SINDy method and find that further nonlinear optimization of the sparsity matrix coefficients improves model performance and limits drift over time. This SINDy-plus-optimization method is able to recover an accurate low-order linear model and can likely be extended to nonlinear process models with some modifications to the forcing functions and selection process for state variables. We also demonstrate that the data-driven methods for creating reduced order models for highly chaotic natural systems may not be adequate, despite modifications to capture system memory. Hence, greater efforts are required to develop appropriate machine learning methods for creating reduced order models of complex natural systems where there is a higher memory in the system. There are two main hindrances to improving the performance of the streamflow model: the low number of data points available for training and testing as well as missing values in what data is available. This lack of data availability/completeness may be addressed through interpolation between data points to generate additional data for use. One such technique that could be utilized is the construction of a cubic smoothing spline. Further, additional nonlinearity in the model functions may be required to reproduce the apparent chaotic behavior of this natural system. This might be achieved through the addition of higher-order polynomials to the function library or even the heaviside function as an operator on inputs and state variables.

Future research would benefit from expanding the scope of state variables included in the process model to include other variables such as temperature and pressure of unit operation blocks and internal flows. Particularly for model applications involving control or observation, a more complete picture of the state space may be required. Incorporating these new state variables into the SINDy method will require greater consideration of the input excitation function frequency and amplitude, and measurement frequency used during data collection to account for differing time-scales among heterogeneous state variables. Additionally, some physics-based a prior knowledge of what function classes will likely appear in the model structure will become more important to limit the possible function space and thus reduce the computational time required to solve the SINDy regression problem.

Acknowledgements This work was supported in part by U.S. National Science Foundation (CBET 1805741). We also thank Dr. Sebastian Oberst and Dr. Merten Stender for their feedback and suggestions for improvement of the models.

Author contributions Material preparation, data collection, simulations and analysis were performed by WF. The first draft

was written by WF. SS conceived the study and all authors co-designed the research approach and study. All authors read and approved the final manuscript.

Funding This work was supported in part by U.S. National Science Foundation (CBET 1805741).

Data availability The datasets generated and analyzed during this study are not available publicly due to the large dataset size and ongoing PhD research of first author, but are available from the first author and/or corresponding authors on reasonable request.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Kroll, A.: Grey-box models: concepts and application. New Front. Comput. Intell. Appl. **57**, 42–51 (2000)
2. Ljung, L.: Black-box models from input-output measurements. In: Imtc 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (cat. No. 01ch 37188), vol. 1, IEEE, pp. 138–146 (2001)
3. London, A.J.: Artificial intelligence and black-box medical decisions: accuracy versus explainability. Hastings Cent. Rep. **49**(1), 15–21 (2019)
4. Al-Malah, K.I.: Aspen Plus: Chemical Engineering Applications. Wiley (2016)
5. Xu, C.-Y., Singh, V.P.: A review on monthly water balance models for water resources investigations. Water Resour. Manage **12**(1), 20–50 (1998)
6. Vandewiele, G., Xu, C.-Y.: Methodology and comparative study of monthly water balance models in Belgium, China and Burma. J. Hydrol. **134**(1–4), 315–347 (1992)
7. Qu, Y., Vogl, G.W.: Estimating dynamic cutting forces of machine tools from measured vibrations using sparse regression with nonlinear function basis. In: Annual Conference of the PHM Society, vol. 13 (2021)
8. Raphaldini, B., Teruya, A.S.W., Brandt, D., Araújo, R.M., Franco, D.R., dos Santos, N.B., da Rocha, R.d.M.: Data-driven low order model of the geomagnetic field during the laschamp excursion
9. Xie, X., Liu, W.K., Gan, Z.: Data-driven discovery of dimensionless numbers and scaling laws from experimental measurements. arXiv preprint arXiv:2111.03583 (2021)
10. Zhang, Y., Duan, J., Jin, Y., Li, Y.: Discovering governing equation from data for multi-stable energy harvester under white noise. Nonlinear Dyn. **106**, 1–12 (2021)
11. Mokhtari, F., Imanpour, A.: Data-driven substructuring technique for pseudo-dynamic hybrid simulation of steel braced frames. arXiv preprint arXiv:2110.02548 (2021)
12. Lakshminarayana, S., Sthapit, S., Maple, C.: Data-driven detection and identification of iot-enabled load-altering attacks in power grids. arXiv preprint arXiv:2110.00667 (2021)
13. Goharoodi, S.K., Dekemele, K., Dupre, L., Loccufier, M., Crevecoeur, G.: Sparse identification of nonlinear duffing oscillator from measurement data. IFAC-Papers OnLine **51**(33), 162–167 (2018)
14. Subramanian, R., Moar, R.R., Singh, S.: White-box machine learning approaches to identify governing equations for overall dynamics of manufacturing systems: A case study on distillation column. Mach. Learn. Appl. **3**, 100014 (2021)
15. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Sparse identification of nonlinear dynamics with control (sindyc). IFAC-PapersOnLine **49**(18), 710–715 (2016)
16. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
17. Stender, M., Oberst, S., Hoffmann, N.: Recovery of differential equations from impulse response time series data for model identification and feature extraction. Vibration **2**(1), 25–46 (2019)
18. Farlessyost, W., Singh, S.: Reduced order dynamical models for complex dynamics in manufacturing and natural systems using machine learning. arXiv preprint arXiv:2110.08313 (2021)
19. Matpan, H.: Data driven model discovery and control of longitudinal missile dynamics. Master's Thesis, Middle East Technical University (2021)
20. Hoffmann, M., Fröhner, C., Noé, F.: Reactive Sindy: Discovering governing reactions from concentration data. J. Chem. Phys. **150**(2), 025101 (2019)
21. Williams, T.: Process dynamics and its application to industrial process design and process control. IFAC Proc. Vol. **1**(2), 595–601 (1963)
22. Rivera, D.E., Lee, H., Braun, M.W., Mittelmann, H.D.:"plant-friendly" system dentification: a challenge for the process industries. IFAC Proceedings Volumes **36**(16), 891–896 (2003)
23. Diasakou, M., Louloudi, A., Papayannakos, N.: Kinetics of the non-catalytic transesterification of soybean oil. Fuel **77**(12), 1297–1302 (1998)
24. Wenzel, B., Tait, M., Módenes, A., Kroumov, A.: Modelling chemical kinetics of soybean oil transesterification process for biodiesel production: an analysis of molar ratio between alcohol and soybean oil temperature changes on the process conversion rate. Int. J. Bioautom. **5**, 13 (2006)
25. Zapata, B.Y.L., Medina, M.A., Gutiérrez, P.Á., de León, H.H., Beltrán, C.G., Gordillo, R.M.: Different approaches for the dynamic model for the production of biodiesel. Chem. Eng. Res. Des. **132**, 536–550 (2018)
26. Thornthwaite, C.W.: An approach toward a rational classification of climate. Geogr. Rev. **38**(1), 55–94 (1948)
27. Thornthwaite, C.W., Mather, J.R.: Instructions and Tables for Computing Potential Evapotranspiration and the Water Balance. Technical Report, Centerton (1957)
28. Alley, W.M.: On the treatment of evapotranspiration, soil moisture accounting, and aquifer recharge in monthly water balance models. Water Resour. Res. **20**(8), 1137–1149 (1984)
29. Makeri, M.U., Karim, R., Abdulkarim, M.S., Ghazali, H.M., Miskandar, M.S., Muhammad, K.: Comparative analysis of

the physico-chemical, thermal, and oxidative properties of winged bean and soybean oils. Int. J. Food Prop. **19**(12), 2769–2787 (2016)
30. Barker, H., Tan, A., Godfrey, K.: The performance of multilevel perturbation signals for nonlinear system identification. IFAC Proc. Vol. **36**(16), 663–668 (2003)
31. Johnston, C., Peverly, J., Soil, V.C., District, W.C.: Watershed Implementation Plan for Lake Vermilion and North Fork Vermilion River, Vermilion County, IL (2008)
32. Bogner, W.C., Hessler, K.E.: Sedimentation survey of lake vermilion, Vermilion county, IL, ISWS Contract Report CR 643 (1999)
33. Abatzoglou, J.T., Brown, T.J.: A comparison of statistical downscaling methods suited for wildfire applications. Int. J. Climatol. **32**(5), 772–780 (2012)
34. USGS: USGS 03338780 NORTH FORK VERMILION RIVER NEAR BISMARCK, IL. https://waterdata.usgs.gov/nwis/uv/?site_no=03338780&agency_cd=USGS. Accessed: 2021-07-15